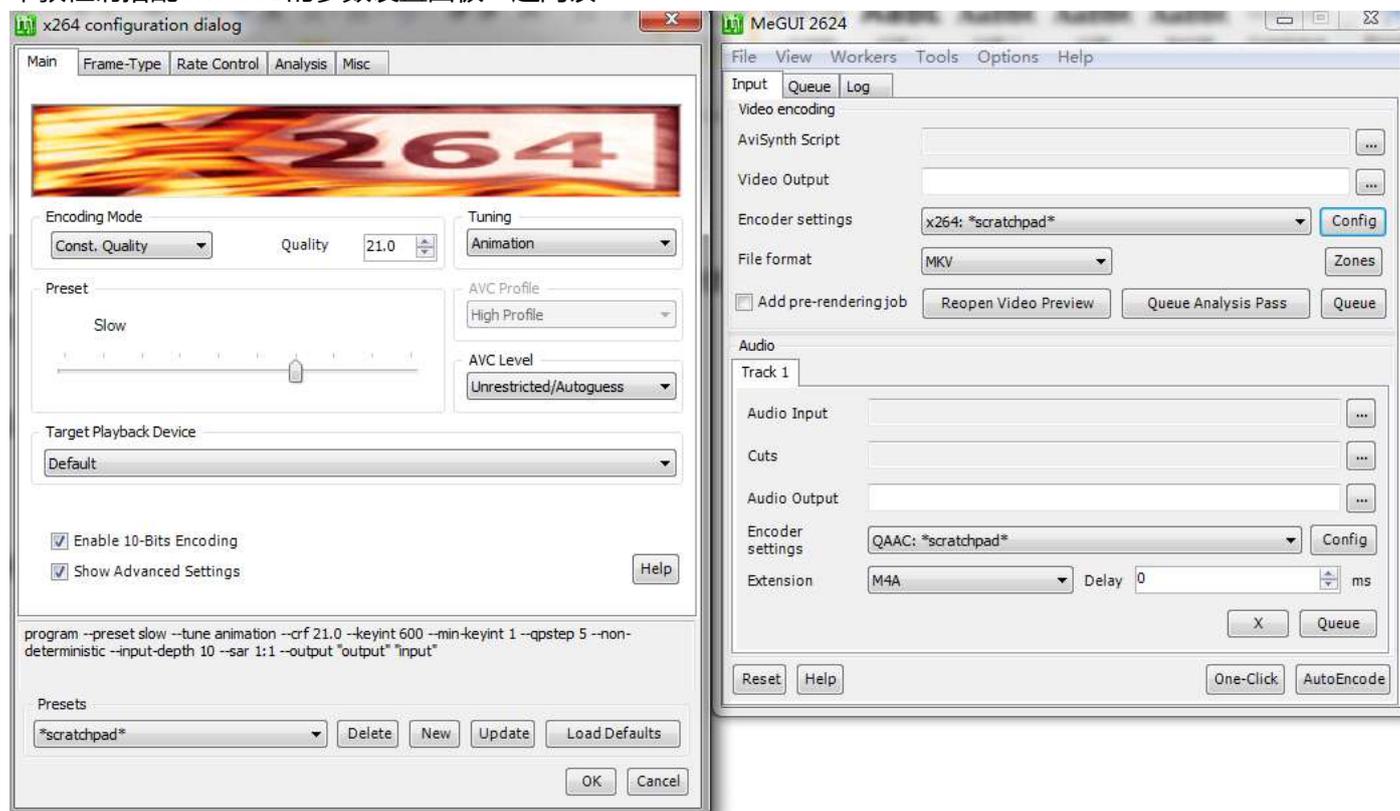


# VCB-Studio 教程 09 x264 参数设置

## 0. 前言

本教程旨在讲述 x264 参数设置的技巧，并侧重于 vcb-s 定位的使用：10bit，动漫，高码率，高参数。对于非这类定位的压制，虽有叙述但不详细。

本教程请搭配 MeGUI 的参数设置面板一起阅读：



本教程会按照标签卡 ( Main->Frame Type->...->Misc ) 分别讲解，每个标签卡内，按照从左上到左下，到右上到右下的顺序，讲述每个参数的使用，并给出它们在批处理命令行中的写法。或者，你完全可以在 MeGUI 中调好，然后复制下方的命令行（去掉开头的 program 和结尾的 --output "output" "input" ）到批处理。

后续会有教程讲述低码率下 8bit 压制的技巧。

**不要将本篇中讲述的任何参数和逻辑套用在 x265 上。**

# 1. Main 标签卡, x264 基础设置

Main 标签卡里面都是最主要的参数,新手上路,掌握 Main 标签卡的内容,就足以去一些普通字幕组担任压制了。

Encoding Mode ( rc , ratecontrol , 码率控制方法 ) :

ABR: Average Bit Rate,指定一个平均码率。x264 会试图让整部视频的平均码率达到你的给定值。如果视频前后编码复杂度相差很大,那么码率的时间分配效果就很差,尤其是到了结尾,为了达到预定的码率值,x264 经常不得不采用过高/过低的码率。所以一般不推荐这个模式。命令行: --bitrate 1000,1000 是码率,单位 Kbps。

Const quantizer : cq 模式,固定量化模式。所有 P 帧(下文有讲)采用一个固定的 Quantizer。Quantizer,量化,是一种衡量图像压缩程度的方法,用 0-69 的浮点数表示,0 为无损。图像被压缩的越多,量化值越大,码率越低,注意量化值不一定代表目视质量,比如说一个纯色的图像可以以很高的量化值被量化,占用的体积很小,而一个很复杂的图像就算量化值不高,但是压缩后观感也可能很差。命令行: --qp 23, 23 是默认值。I 帧和 B 帧的平均 quantizer 由 ipratio 和 pbratio 决定,这个后续教程会说。因为 quantizer 不能够较好的体现质量,所以这个模式一般由下文的 Const Quality 模式替代。就算不用 cq 模式,quantizer (量化)这个概念依旧在编码中存在,只不过编码器可以智能的浮动 qp 值。

Const Quality/Const Ratefactor , crf 模式,固定质量模式。x264 用一种结合人心理学,估算出来的值,来衡量视频的目测质量,这就是 rf (ratefactor),用浮点数表示,0 为无损,越高质量越差。crf 就是在视频前后采用恒定的 rf,从而使得视频前后的目测质量几乎一致。crf 模式下,码率的时间分配效果是最理想的,也是最常用的模式。

一般人常见的视频,crf 设置在 18-26 左右。通常采用 19-21.5 就能使得 Rip 看上去很不错。如果需要绝对好的质量可以降低到 16,但是码率也会很高。注意,crf 在搭配不同的参数前提下,实际造成的目测效果还是有差距的,甚至可能很大,因此不能一概的认为 crf 代表目视质量。命令行: --crf 23,23 是默认值。

一般 vcb-s 用的 10bit 1080p BDRip , crf 选择在 16~18 之间。

2pass/3pass 模式:也是指定平均码率,不过通过多次编码(第一次预编码一般是以很快的速度)来先衡量视频前后编码复杂度,从而优化码率的时间分配。前面编码输出 stats 文件表示记录,后面编码读取 stats 文件以得知前后编码复杂度。如果对码率/体积有要求,一般更推荐花更多时间走 2pass 模式,而不是直接用 abr,也不要去看 crf 碰运气希望出来的体积正好达到要求。Npass 的终极目的是,让出来的视频接近 crf 模式。所以对体积没有要求的前提下,优先使用 crf 模式而不是 Npass。

Preset (效率预设):

效率预设值。在编码效率和编码速度(时间)之间进行取舍。越往右滑越慢。最快的 ultrafast 速度媲美 GPU 加速,最慢的 placebo 所需的时间可以是 ultrafast 的近百倍。Preset 越高,x264 的码率效率越高,意味着单位码率能做到的画质越好。命令行: --preset "slower"

采用你最大能忍受的预设值,一般推荐在 slow/slower/veryslow 三档。x264 会自动设置很多参数来调节你对速度与效率的取舍。vcb-s 所用的参数一般基于--preset "veryslow"

Tuning (画质预设)

画面预设值。告诉 x264 参数应该为哪种片源优化:

None : 默认设置, 适合一般/混合类的片源;

Animation : 动漫, 动画 ( 虽然实际高质量动漫 BDRip **极度不推荐** )

Film : 电影。通常演唱会、非动漫特典, 都采用这个预设。

Grain : 在噪点明显的源上, 致力于保留噪点。注意这个模式在搭配 crf, 尤其是低 crf, 体积惊人, 因为花费了很多码率来保留噪点。tune grain 往往比较适合极高码率的编码, 因为其码率分配策略优先是细节保留, 而不是编码效率。

still image : 静态画面, 适合那种例如相册类的片源

psnr/ssim : psnr 和 ssim 是常用的两种衡量源和成品画质区别的客观评测跑分, 然而高分的画面并不一定目视效果优良。x264 的默认参数是为了目视效果而优化, 不是为了跑分而优化。使用这两个, x264 会将画面为了跑分优化, 代价是牺牲目视质量。一般当且仅当你真的是拿去跑分对比的时候才用

命令行: --tune film

preset 和 tune, 这两个参数是官方经过大量实验和优化之后, 设定的调整选项。如果你对更详细的参数不了解, 或者对某种片源怎么设置不知道, 或者不知道如何调节参数权衡速度与效率, 使用官方的参数, 比你自已瞎调一般都要保险:

编码一部电影, 采用折中的质量和折中的速度:

```
--crf=23.0 --preset medium --tune film
```

为了不超过 1000K 上传 b 站, 编码一部动漫, 2pass 模式, 码率在 920 以内 ( 留 80 给音轨 ), 并且不惜龟速最大化编码效率 ( MeGUI 中选 automated 2pass 920KBps, 下文是命令行写法 ):

```
--bitrate 920 --preset placebo --tune animation --pass 1 --stats "temp.stats"
```

```
--bitrate 920 --preset placebo --tune animation --pass 2 --stats "temp.stats"
```

编码一个基本静止相册视频, 并采用绝对优秀的画质, 和比较慢的速度:

```
--crf=16.0 --preset slower --tune stillimage
```

AVC Profile : 设定 H.264 的 Profile ( 算是编码高级程度 ), 一般现在设备都支持 High Profile 了。所以选 High 就行; 10bit 编码则强制为 High 10 Profile。一般不用在命令行中设置; x264 会自动选择合适的。

AVC Level : 跟 profile 相似, 但是分的更细, 一般现在支持性都比较好了, 选 unrestricted/autoguess, x264 会自动设置最好的值。

Target playback device ( 目标播放设备 ):

如果你在移动设备上播放, 那么首先保证你使用 8bit x264 编码(不要勾选 enable 10bits encoding), 然后可以用这个选项来设置播放设备, x264 会自动限制 profile/level 和更多其他参数 ( 代价是牺牲一点压缩率)。绝大多数的情况下, 它给你的参数过于保守, 极少数的情况下还是超过了设备限制, 但是在你没有详细实验, 或者没有其他可靠的教程下, 使用这个选项是最好的选择。

Enable 10bits encoding, 这个选项开启了, MeGUI 会使用 10bit x264 编码 ( 调用 MeGUI\tools\x264\_10b 下的内容 )。出来的成品是 10bit 的。

## 2. IPB 相关的知识

在讲述 x264 进阶知识之前，有必要先说说 IPB 帧相关的知识点。这部分内容非常基础，从播放到压制，很多事情的解释和研究都离不开它。

### 2.1、IBP 帧

视频压缩的过程中，对于一段时间内相似的帧，采用记录第一张，后续几张只记录和第一张的区别，这种想法是很自然的。由此引申出两种帧：

I 帧(Independent Frame)，独立编码的帧。I 帧相当于记录一张 jpg；一般常见于一个场景开头。后续的帧就需要依赖 I 帧；

P 帧(Predictive Frame)，需要依赖其他帧来编码的类型。P 帧需要参照在它之前的 I 帧或者其他的 P 帧，因为它只记录差别，所以对于那种前后差别很小甚至没有的帧，使用 P 帧编码能极大地减少体积。

MPEG2 之后，引入了第三种帧：

B 帧(Bi-directional Frame)，双向预测帧。B 帧跟 P 帧类似，都是需要参照别的帧，区别在于 B 帧需要参照它后面的帧，B 帧的记录和预测方式也做了调整，使得 B 帧的预测方式更灵活，对付高度静态规律的场景可以更有效的降低体积。

一种典型的排列方式：

IPBBPBPIPPB.....

视频开头一定是一个 I 帧。

## 2、IDR 帧与 GOP 区间

I 帧中，有一类特殊的 I 帧，叫做 IDR 帧。IDR 帧的性质是，比如第 1000 帧是 IDR 帧，那么这一帧相当于一个分水岭，从 1001 帧开始，所有的帧都不能再参照 1000 帧之前的帧。在 closed GOP 规定下(x264 设置中可以指定，并且 vcb-s 一直指定)，0~999 帧也不允许参照这个 IDR 帧以及之后的帧。等于说 IDR 帧将视频分割成两个独立的部分：前面的 (closed GOP 规定下) 不能参照后面的，后面的不能参照前面的。

这个性质在播放的时候额外有用：如果我直接从第 1000 帧开始播放，我可以毫无问题的播放下去，因为我不需要参照 1000 帧之前的内容完成解码。我从开头播放，直到 999 帧的时候，我都不需要参照 1000 帧及它后面的东西；1000 帧之后的数据都损坏了，0~999 帧也能正常播放。

IDR 的全称叫做 Instantaneous Decoder Refresh，意思是，解码到当前帧，解码器就可以把缓存全清了——之前的所有帧信息都没用了；后续帧不会再去参照它们。

视频开头的 I 帧一定是 IDR 帧。

有时候，我们用 I 帧表示 IDR 帧，i 帧表示非 IDR 的 I 帧。这种场合下，I 帧和 i 帧都是 independent frame，区别

在是否是 IDR。

两个 IDR 帧之间的区间,从一个 IDR 帧开始,到下一个 IDR 前的帧结束,叫做 IDR 区间,又叫做 GOP 区间。closed GOP 设定下,GOP 区间可以看做是独立的一段视频:它里面的所有帧,都不需要参照任何区间之外的东西,只要一个 GOP 区间是齐全的,区间里面所有的帧都能被解码。我们平时看的视频就是多段 GOP 区间连接起来的。

在我们拖动进度条的时候,为啥有时候会卡顿,或者拖不准呢?其实是播放器干了这些事:

1. 计算你拖动进度条的时间,找出它应该是哪一帧,假设为 N;
2. 通过搜索视频索引信息,找出 N 帧前面最近的一个 IDR 帧,假设为 M( $M \leq N$ )。很显然,M 和 N 同属于一个 GOP 区间,这个区间开头的帧是 M;
3. 如果你的播放器设置了快速索引,视频将直接从 M 帧开始播放,因为 M 帧是 IDR,它不需要参照任何帧,所以立刻可以开始播放。这是为啥你会发现开始的地方总是在之前一点;
4. 否则,如果你的播放器设置了精确索引,解码器会从 M 帧一直开始解码,解码到 N 帧,然后显示 N 帧的画面,并继续播放。

当 N-M 比较大的时候,播放器可能需要先解码几百甚至上千帧,才能继续播放。如果视频允许很长的 GOP 区间,这个视频在播放的时候,拖动进度条就特容易卡顿。

在 x264 中,如果设置了--min-keyint 1,那么所有 I 帧都是 IDR 帧(vcb-s 也一直在用)。而 GOP 区间最大的范围是通过--keyint 指定的。下一章我们就将讲述这些知识。

### 3. Frame Type 标签卡，帧类型参数

deblocking 这块其实没啥重要的，基本是保持默认的节奏。

Deblocking (去色块)：这是 H.264 的一个固有的性质。现代编码器都是基于 MacroBlock 编码，把画面划分为一块块的区域，有损编码后，区域之间就可能出现可见的“分界线”。Deblock 的作用就是去除它们。开启是肯定的，然后下面两个参数分别是 deblock 的 alpha 和 beta 参数。这两个参数默认是 0:0。一般认为合理的范围是 -3:-3 ~ 3:3 之间。越是高码率的压片越建议调低；因为设置高了，deblock 的强度会变大，对画面有模糊的效果。通常设置在 -1:-1 ~ 1:1 之间就行了。默认 0:0 已经算是比较合理的选择。

deblock 的强度和 qp 值有关，所以 deblock 设置不变时，码率越高 (qp 值相对越低)，deblock 的强度就会相应降低 (x264 自身的逻辑也是高码率降低 deblock 的强度)。命令行：`--deblock -1:-1`

CABAC：勾选就行了。x264 编码的最后一步相当于做了一个数据的无损压缩 (类似 zip/rar/7z, 这也是为啥视频丢到压缩软件里压根没有任何压缩空间)，压缩算法可选 CABAC (Context Adaptive Binary Arithmetic Coding) 和 CAVLC (Context Adaptive Variable Length Coding)，通常 CABAC 压缩率更高，编解码也更慢，但个别情况下 CAVLC 压缩率会更高 (例如单色图像)。默认就是 CABAC，命令行不用改。

Maximum GOP Size, `--keyint`，规定了最远两个 IDR 帧之间的距离。MeGUI 中 0 表示无限，命令行中则需要指定为 `infinite`。这个值越大，给编码器灵活度越高，因为编码器不用被迫在不需要的情况下插入 IDR 帧；但是也意味着拖动进度条的时候压力越大。特别是 `keyint=inf` 的时候，一些真人访谈场景，如果不切换场景，很可能从头到尾只有一个第 0 帧是 IDR。一般来说大家设置在 250~900 之间。对于在线视频，360~480 是不错的选择。本地视频，480~720 会比较合适。

Minimum GOP Size, `--min-keyint`，规定了最近两个 IDR 帧之间的距离。一般编码器在检测到场景转换的时候，会插入 I 帧。如果新插入的 I 帧距离上个 IDR 帧的距离  $\geq \text{min-keyint}$ ，编码器就会将它设置为 IDR 帧。否则会插入非 IDR 的 i 帧。将 `min-keyint` 设置成 1，所有 I 帧都会变为 IDR 帧。一般选择 `--min-keyint=1`  
i 帧和 IDR 帧都只进行 intra prediction (所有参照信息都是帧内，不借助前后帧)。区别在于 i 帧的前后 P/B 帧可以互相参考，而 I 帧不允许。

关于 `min-keyint` 的设置有一种说法是考虑到画面中出现反复闪烁时，设置一定大小的 `min-keyint` 可以避免插入 I 帧，而是插入 i 帧。但实际上，x264 的 scene change 检测是十分智能的，如果前后的场景可以有互相参考的价值，那么即便中间有几个突变帧，也不会贸然地插入 I 帧。实际上，这种情况在闪烁的地方插入 P 帧是完全没问题的，因为即便是 P 帧，其中的各个 macro-block 依然可以作为 I Block (独立编码)，所以没有使用 i 帧的必要性。

(P 帧和 B 帧都可以有 Intra Block，表现为某个 macroblock 不借助前后帧进行参考。比如说在运动场景中突然出现了一个小的新物体，整个帧可以设置为 P/B 帧参照前后，但是新物体的部分则采用 Intra Block 编码，因为没有可以参照的。)

事实上插入 i 帧还会导致兼容性问题，例如为了 blu-ray compat 就需要设为 1，所以始终用 `min-keyint 1` 是最佳的选择。

Open GOP。如果开启，那么前一个 GOP 的 B 帧将可以参照下一个 GOP 的里面的帧。GOP 规定后面的不允许参

照前面的，但是前面的能否参照后面的，则是由 OpenGOP 决定的。如果开启，那么在特定场景下可以增加编码效率，但是一些播放设备和播放器不支持。一般来说开启与否问题都不大。

Slicing 那块是给编码蓝光原盘用的。日常编码永远用不到；永远。

Bframes 部分是给你设置 B 帧的。

Weighted Prediction for B frame，允许 B 帧的加权预测。百利无一害的选项，所以开启。默认就是开的，命令行：`--weightb`

Number of B-Frames，最大允许连续 B 帧数量。这个值越大，编码时间稍有提高，对压缩率也有帮助。一般认为真人电影设置为 3~8，动漫设置为 6~12 左右比较合适。

Adaptive B-Frames，采用什么算法来决定是否采用 B 帧。0-off 表示永远能用就用，1-fast 是快速算法，2-normal 采用的是常规算法。2 比较优秀，也会较慢。一般推荐 `--preset slow` 及以上时候使用 2。命令行：`--b-adapt 2`

B-Pyramid，B 帧参照其他 B 帧的方式。Disable 表示不允许 B 帧参照其他的 B 帧，Strict 为了 BD 原盘播放兼容性而设置，normal 则不加限制。一般选择 normal 以最大化编码效率。命令行：`--b-pyramid normal`

Others 部分主要是 ref (参考帧) 的设置

Number of Reference Frames，即 `--ref`。表示某一帧最多允许参照多少其他的帧。ref 对编码所需时间的影响是线性的，ref 能带来的收益则是非线性的。也即越高的 ref，额外收益越低。ref 是一个以编码时间换取编码效率的参数。

ref 的数量不仅影响编码效率，而且影响解码效率，以及解码时最大缓存帧数。ref 和分辨率与帧率相乘后决定了播放设备需要达到的解码能力，H264 各个 Level 对其 ref 上限的支持有相应的限制。

以日常的动漫，一般来说，ref 设置如下：

追求速度的编码可以用 `ref=4`。速度会很理想。

日常编码，ref 可以选择 6，相对 `ref=4` 可以节省 2%~4% 左右的体积。

追求压缩比的编码，ref 可以选择 8，相对 `ref=4` 可以节省 4%~8% 的体积。

追求极致压缩比的编码，ref 可以选择 13 左右。相对 `ref=4` 可以节省 6%~15% 的体积。

特别静态的场景，或者特别动态的场景（比如噪点多的真人电影），ref 开高的意义均不大。反而是动态程度一般，场景混杂的，开高 ref 效果比较好。

Number of Extra I-frames，即 `--scene-cut`。这是让 x264 决定是否切换场景的灵敏度。x264 会在场景切换的时候插入 I 帧，这个参数越低，x264 越容易判断场景做了切换，越鼓励插入 I 帧。通常保持默认就好。PS：网上教程多半为越高越鼓励 I 帧，然而实测却相反——不管怎样，默认值已经不错，所以一般也不用纠结自己去改。

P-Frame Weighed Prediction: P 帧的加权预测方式。一般选择 smart (默认)。

Interlace mode 和 pulldown 选择 none。这是为电视编码才用得到的选项。

Adaptive I-Frame Decision，这是采用 `--scene-cut` 决定智能插入 I 帧的前提。开启它。

## 4. Rate Control 标签卡/码率控制

rate control 标签卡里面的参数，大多是能在不增加减少编码时间的前提下，大幅影响码率分配的参数，即如何让参数分配的合理。

quantizer 部分是给你设置 qp 值的。

Min/Max/Delta: 每一帧用的 qp 最小值，最大值，相邻两帧间最大允许的 qp 差值。一般保持默认就好了。在定位高还原的编码中，如果想防止某一帧被压的很烂，可以设置--qpmax 36 左右。

Quantizers Ratio(I:P:P:B): 决定 IPB 三帧质量削减的比例。这两个比例越大，IPB 三种帧之间画质差异度越大，反之亦然。更大的差异度有助于发挥三种帧的定位特性，但是也会使得画质两极分化严重。默认值是很好的选择。

Chroma QP offset: 这个参数是调整 Luma 平面和 Chroma 平面间，qp 值差异的一个参数。因为我们说过，这两个平面的重要性是不等同的，所以可以考虑将 chroma 编码的烂一些，所以 444 采样的片子(即 Luma 和 Chroma 都是全分辨率记录)，x264 自动加上 5 的 offset——让 chroma 的 qp 值比 luma 平均多 5。而对于 420/422 采样的片子，有时候 Luma 已经被削减之后，有必要在编码的时候额外保护，就可以通过负值的 offset 让它编码效果更好。一般维持默认，或者 422 片源自己手动+2。命令行：--chroma-qp-offset 2

MediaInfo 中看信息，往往这个值跟你手动设置的并不一样；第一个 x264 对于 444 采样会自动+5，第二个后文的 psy-trellis 选项也会调整。

Credits Quantizer，这个不用改。

VBV Buffer Size 和 VBV Maximum Rate 是用来限制码率上限的，一般不需要改动。如果你希望设置一个码率上限，比如说 20Mbps，你就可以将 VBV Maximum Rate 设置为略低一点(比如 19000)，VBV Buffer Size 设置为 20000

VBV initial Buffer: 不用改

Bitrate Variance，这是在 abr/2pass/3pass (统称为 N Pass)中，规定最终输出的成品，和原来指定的码率允许有多少误差。单位是%。比如指定 1.0，意味着码率可以相差 1%。这个值越小，N pass 最终输出的码率准确度就越高，但是意味着到了编码的最后阶段，码率需要不合实际的去调整以求总码率尽可能的接近预设值。

Quantizer Compression，--qcomp，这个参数决定了 qp 的时域变化灵活度。越低的数值代表灵活度越高，qp 值变动越大，效果就是高动态场景下烂的比较严重，因为 x264 会倾向于提高高动态下的 qp 值(特别是引入 mbtree 之后)。通常认为，越是倾向于高画质编码的，--qcomp 需要给的越高，反之亦然。qcomp=0 的时候效果接近固定比特率，qcomp=1 的时候效果接近固定 qp 值。qcomp 的作用会受到 mbtree 的影响。这个我们下文细谈。

Temp. Blur 的两个参数不用去多管，不会用到的。

Adaptive Quantizers，简称 AQ。没有 AQ，x264 会倾向于在平面和纹理处降低码率。造成的效果就是线条部分看上去还行，但是平面大幅度 block，纹理烂掉。AQ 的作用就是来防止码率在纹理和平面处被过分的降低。

AQ Mode: 选择 AQ 的算法。Disable(aq-mode=0,不用 aq), Variance AQ(aq-mode=1)和 Auto-Variance AQ(aq-mode=2)。tMod 版 x264 还加了 mode=3/4/5, 只不过在 megui 中需要用自定义命令行来启用。

一般来说, mode=3 最好, 最适合动漫, 但是码率稍高; mode=1 效果中等, 比较安全, 不容易出现教烂的帧。mode=2 比较省码率, 但是偶尔容易出现烂帧。mode=4 是由 2 优化而来, 更保险一点。

一般来说, 日常压动漫就选 3, 压真人特典选 1 或者 4

Strength, 就是 aq 的强度。动漫选 0.6~1.0 左右, 真人选 0.8~1.2 左右。越是高质量的编码, aq 的 strength 应该越高。

Quantizer Matrices, 无视就好了。

MB-tree(Marco-Block Tree)是 x264 后期引入的一种码率控制和决策的工具, 在时域和空域都有重要的影响。mbtree 的原理简单点说, 就是在编码过程中, 被大量参照的 block(被前后帧参照, 或者被同一帧其他部分参照), 给的 qp 值降低, 画质更好, 体积更大; 反之, 被参照少的帧, qp 增加, 画质更烂, 体积更小。其逻辑在于, 被参照多的 block 理应有更好的画质, 这可以让参照它的更多 block 受益。mbtree 的弊端主要在通常将平面和纹理涂抹的较为过分(这些 block 通常是直接参照别的 block), mbtree 也倾向于降低高动态部分的质量。如果关闭 mbtree, 通常细节会更好, 但是线条会有些欠码。

普遍认为, 在中低码率的编码中(crf>18), mbtree 永远是利大于弊的。在极高码率的编码中(crf<16), mbtree 永远是弊大于利的。在高码率(crf=16~18), 越是动态多, 噪点多, 线条锐利的片子, mbtree 的正面作用越强。

开启 mbtree(--mbtree 为开启, --no-mbtree 为关闭), 体积会减少很多。所以一般开 mbtree, crf 可以相比于关 mbtree 低 1.0 左右。比如 crf=16.5 mbtree=1/crf=17.5 mbtree=0 是 vcb-s 常用的选择。

mbtree 开启的时候, qcomp 的灵活度会被放大。所以一般开 mbtree, 需要增加 qcomp (qcomp 数值增加, 灵活度降低)。关闭 mbtree 的时候, qcomp 的敏感度则降低。比如 vcb-s 常用的参数:

crf=16.0 mbtree=1 qcomp=0.80/crf=17.0 mbtree=0 qcomp=0.70

crf=16.5 mbtree=1 qcomp=0.75/crf=17.5 mbtree=0 qcomp=0.60

crf=17.0 mbtree=1 qcomp=0.75/crf=18.0 mbtree=0 qcomp=0.60

crf=19.0 mbtree=1 qcomp=0.70/(crf=19 的时候不宜关闭 mbtree)

Nb of Frames for Look-ahead, 这个一般设置为帧率的 2~3 倍就好。比如 24p 的设置为 70。

## 5. Analysis 标签卡/分析

analysis 的标签卡里面则是 Motion-Estimation(动态分析)相关

Motion Estimation, 原则上是将每一帧分成 blocks (比如 8x8 的块), 分析每一个 block 在时间变化中, 运动的方向和距离, 和运动前后的区别。通过这些 ME 分析, 从而实现 PB 帧对于动态场景的良好适应。

Chroma ME, 是否对 Chroma 平面启用 ME 分析。肯定是要的。

ME Range(--me\_range), 表示搜索的最大范围。一般 720p 给 16~24,1080p 给 24~32。这个值越大速度越慢。

ME Algorithm(--me):动态搜索的算法

Diamond(--me dia), 菱形搜索, 不推荐

Hex(--me hex), 六边形搜索, 只在非常追求速度的时候推荐

Multi Hex(--me umh), 多重六边形搜索。非常均衡的算法, 效果非常好, 速度适中。推荐一般情况下使用

Exhaustive(--me esa), 全面搜索。很慢.....

SATD Exhaustive(--me tesa) 经过转换的全面搜索。速度比 esa 稍微慢一点点, 但是效果好一点, 所以一般有时间浪费的都选 tesa 而不是 esa。

tesa 比起 umh, 提升在 5%~10%左右。时间上则是 150%~200%的提升.....

Subpixel Refinement (次像素级别的优化, --subme)

从 0~11 效果越来越好, 速度也越来越慢。追求速度推荐用 7, 追求质量推荐用 10。其他的其实都没啥好选的...

me/subme/me\_range 三个选项, 共同组成 ME 的参数组合。ME 是以编码时间换效率的另一套参数 (之前是 --ref/--bframes 构成的 Frame)。一般来说, 动态不高的番, 应该用较强的 frame 参数, 动态高的番则是用较强的 ME 参数。

一般来说, 下面三个组合是很好的快中慢三种选择

```
--me umh --subme 7 --me_range 16
```

```
--me umh --subme 10 --me_range 24
```

```
--me tesa --subme 10 --me_range 24
```

MV Prediction Mode 和 trellis 一般默认就好。

Psy-rd Strength(--psy-rd): 心理学优化强度。psy 相关是一种 x264 引入的主观优化: 在欠码的时候, 人眼宁愿看到失真, 也不愿看到大范围的模糊。虽然这种失真对客观的还原度来说不利, 但是它有利于保留画面纹理, 编码前后的图像看上去违和感较低, 细节锐度较好。

psy-rd 就是决定这种强度的参数。一般压制动漫选择 0.4~1.0, 压制真人选择 0.7~1.3。越是高质量的编码, 可以开的越高。但是在 crf 模式下, 开高了 psy-rd 也会提高码率。

psy-trellis, 在 psy-rd 的基础上, 微调保留的噪点之类的微小细节的保留度。一般开 mbtree 时候, psy-trellis 可以给 0.1 左右。mbtree=0 的时候推荐不开。psy-trellis 使用的越高, 会相应的调低--chroma-qp-offset 以求让 chroma 平面的细节得以保留。所以在关闭 mbtree 的时候, 一方面将--psy-trellis 设置为 0, 一方面将--chroma-qp-offset 往下调 1 左右。

命令行: --psy-rd 0.6:0.15

No Mixed Reference Frame, No DCT Decimation 一般保持默认不勾选。

No Fast P Skip: --fast-pskip 速度换质量的参数。高速编码一般启用, 否则不启用: --no-fast-pskip。在 MeGUI 中, 勾选是不启用 ( --no-fast-pskip ), 不勾选是启用(--fast-pskip)

No Psychological Enhancement 禁用 psy 优化, 以追求数据上的高还原。这当且仅当你是为了跑数据的时候才开启。( 比如--tune psnr )

NoiseReduction, x264 的自带降噪。选 0 就好了, 降噪的话, avs/vs 比它好使的多

MacroBlocks 决定启用哪些分块机制。一般都选 all( 默认 )。部分 Profile 和 Level 可能有限制( 比如说 Main Profile 就不能启用全部 ), 这在开启了 Profile 限定之后会自动处理。

Blu-Ray info 不用管。编码蓝光才需要的。

## 6. Misc(miscellaneous , 意思是闲杂/其它)选项卡

Custom Command Line , 是给你自定义的地方。这里写的东西就是前面的补充。比如说:

```
--input-depth 16(输入为 interleaved 16bit)
--aq-mode 3(使用 aq-mode 3)
--log-file "xxx.log" --log-file-level debug (记录压制过程)
```

x264 的使用里面, crf 和 2pass 的取舍有点困难。前者效率高, 只跑一次, 效果好; 后者能实现对码率的精确调控, 但是需要跑两次。在对体积有一定的要求这种场合下, 一种常见的策略是: 1pass 用 crf 跑, 如果体积跑出来差很小, 直接作为最后成品; 如果差不多, 再用 2pass。如果差的太大, 证明用预定的码率很难达到预定的质量, 可能就要重新考虑制作方法。这种就可以用 MeGUI 实现

1pass 的时候, Main 标签卡选 Const Quality, 设置好一个大概的 crf。然后在 custom command line 里面加上 (分别是表示: 这是 1pass / 开启 slow-firstpass, 否则 1pass 参数会自动缩水 / 1pass 的分析记录保留在 xxx.stats 中, 一般在 megui 的根目录下):

```
--pass 1 --slow-firstpass --stats "xxx.stats"
```

1pass 完全是按照 crf 的在跑。如果跑出来码率偏差比较小, 就不必再跑一次 2pass。否则, 2pass 中, Main 里面设置 2pass-2nd pass, 设置你想要的码率。custom command line 里面加上(表示记录文件从 xxx.stats 里面读取):

```
--stats "xxx.stats"
```

不用写--pass 2.因为 Main 里面设置了 2pass-2nd pass 和码率之后, megui 自动帮你设置好--pass 2

Files 不用设置。

UVI 主要是颜色方面的设置。这部分需要结合 avs 后续的教程, 说到不同颜色空间转换时候才需要。目前你可以全部保持默认 ( auto/undefined )

Other 里面基本上不需要勾选。唯一可能需要考虑的是 threads。它规定了 x264 能使用的线程数。一般 threads 不要超过 16; 超过后对画质有一定的负面影响。

默认 0, 让 x264 自己计算。计算的方式是你的框框数量\*1.5。也就是 4C8T 的是 threads=12. 6C12T 则是 threads=18。所以 3930K/4930K 以上最好就要手动设置了。

Input/Output:

psnr/ssim 这两个是两种计算视频编码还原度的测量值。如果勾选, 视频在编码的时候会做计算。一般不需要。开启 psnr/ssim, 如果你没有用对应的--tune, 或者开启了 psy, x264 会警告你, 目前画面没有为 psnr/ssim 优化, 参照它们可能没有意义。

Swiathable : 勾选了之后, x264 会保证你能合并视频。但是其实不勾选也能合并.....

Force SAR 是用来设置 SAR 的。要知道它的作用, 先来了解一下术语 (下文都是 MPEG4 时代的术语, 注意了, 不要跟 MPEG2 时代的混淆):

DAR = Display Aspect Ratio, 播放时候看到的横纵比。如果你看到的播放效果是 16:9 的, 那么 DAR=16:9。  
(MPEG2 时代也有这个概念)

FAR = Frame Aspect Ratio, 每一帧的横纵比。比如如果视频是 640x480 的, 那么 FAR=640/480=4:3。(MPEG2 时代它叫做 Storage Aspect Ratio, 储存恒总比, 简称 SAR。注意这个 SAR 跟下文的 Sample Aspect Ratio 是不同的概念)

SAR = Sample Aspect Ratio, 每个像素的横纵比。如果一个 640x480 的视频, 它的像素恒总比是 1:1, 那么它在播放的时候屏幕显示是 4:3 的。但是你也可以指定它的 sar=4:3, 这样显示的时候, 每个像素相当于是一个 4:3 的长方形。那么整个画面就会再被拉长到 16:9。(MPEG2 时代叫做 Pixel Aspect Ratio, 像素恒总比, 简称 PAR。MPEG2 时代也有个 SAR 的简称, 但是它表示的是 MPEG4 时代的 PAR)

MPEG4 时代, DAR=FAR\*SAR

MPEG2 时代, DAR=SAR\*PAR

两个公式上下对应的简称是完全等价的, 但是两个 SAR 表示的意义是不同的。所以你爬文的时候, 碰到 SAR 的简称, 一定要注意区分, 它是表示视频的物理分辨率, 还是一个像素在播放时候被指定的分辨率。

sar 在编码 DVD 的时候经常碰到。比如说日本的 DVD 多为 NTSC 制式, 它的物理分辨率多为 720x480, 但是播放的时候显示 16:9, 所以:

$$FAR=720/480=3/2$$

$$DAR=16/9$$

$$\text{算得 } SAR=DAR/FAR = (16/9)/(3/2)=32:27$$

那么在编码 DVD 的时候就需要手动设置 sar 为 32:27: --sar 32:27

## 7. VCB-Studio 常见编码参数的解读

vcb-s 的参数一般有两套模板，开 mbtree 的一套和不开 mbtree 的一套。

开 mbtree 的：

```
--preset veryslow --crf 16.5 --threads 16 --deblock -1:-1 --keyint 600 --min-keyint 1 --bframes 8 --ref 13
--qcomp 0.75 --rc-lookahead 70 --aq-strength 0.8 --me tesa --psy-rd 0.6:0.15 --no-fast-pskip
--colormatrix bt709 --aq-mode 3 --input-depth 16
```

解析：

--preset veryslow 先给定预设值是 veryslow。veryslow 会自动给你设置好很多追求质量的编码下，常用的参数。比如 bframes=8 ref=16 --me=umh subme=10 me\_range=24 等。

--crf 16.5，开 mbtree 下高质量的 crf

--threads 16，最多允许 16 个线程。一般是对 8 个框框以上的时候设置

--deblock -1:-1 高质量编码时候常用的 deblock 值

--keyint 600 最多允许两个 IDR 帧的间距为 600 帧，24p 的话就是大概 15s

--min-keyint 1 最小间距 1

--bframes 8 算是较高的值（veryslow 的预设）

--ref 13 预设给了 16，但是一般 10~13 就足够了

--qcomp 0.75。默认是 0.6，但是高质量编码，又是开 mbtree，需要调高。一般调高不超过 0.8 为适

--rc-lookahead 70。设置为帧率的 3 倍，差不多是 70

--aq-mode 3 --aq-strength 0.8 这是广泛认为最适合动漫的 aq 参数组合之一

--psy-rd 0.6:0.15 也是针对开 mbtree 的高画质编码下，很平衡的 psy 参数

--me tesa 从 veryslow 默认的--me umh 提高而来，加强动态的分析(其余的 me 参数保持 veryslow 默认的 --subme 10 --me\_range 24)

--no-fast-pskip 等效于勾选 No Fast P Skip ( Analysis 标签下 )。

--colormatrix bt709 蓝光原盘一般都是这样

--input-depth 16 一般编码我们都是使用 interleaved 16bit 输入

关 mbtree 的：

```
--preset veryslow --crf 17.5 --threads 16 --deblock -1:-1 --keyint 600 --min-keyint 1 --bframes 8 --ref 13
--qcomp 0.6 --no-mbtree --rc-lookahead 70 --aq-strength 0.8 --me tesa --psy-rd 0.6:0.0
--chroma-qp-offset -1 --no-fast-pskip --colormatrix bt709 --aq-mode 3 --input-depth 16
```

解析(与开 mbtree 不同的)：

--crf 17.5 关 mbtree 下,crf 应该适度增加

--qcomp 0.6 关 mbtee 下，一般不需要调节 qcomp，或者仅仅提高一点点，不要超过 0.7

--psy-rd 0.6:0.0 关 mbtree 下，编码器本身会倾向于保留更多噪点级别的细节。psy-trellis 保持 0 就好了（开

mbtree 下是 0.15 )

--no-mbtree 不开启 mbtree

--chroma-qp-offset -1 在 psy-trellis=0 的时候，手动调低这个以保证 chroma 平面画质更好点。

有些时候，当源本身噪点较重，vcb-s 会采用--fgo 1 这个选项。这个选项效果跟--trellis 类似，但是逻辑是更暴力的强制保留噪点。官方 x264 因为觉得这个逻辑无助于提升压缩率，且 trellis 的表现更具有普适性，所以移除了 fgo。但是诸如 tMod 和 kMod 的各种魔改 x264 中依旧有。当然，这个选项也不是天上掉的馅饼，通常需要搭配更高的 crf 来控制体积，且可能造成线条和强纹理部分欠码。